実習 - SSH を使用したネットワーク デバイスへのアクセス

トポロジ

アドレッシング テーブル

デバイス	インターフェイス	IP アドレス	サブネット マスク	デフォルト ゲートウェイ
R1	G0/1	192.168.1.1	255.255.255.0	該当なし
S1	VLAN 1	192.168.1.11	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1

目的

パート 1：デバイスの基本設定を構成する

パート 2：ルータを SSH アクセス用に設定する

パート 3：Wireshark を使用して Telnet セッションを調べる

パート 4：Wireshark を使用して SSH セッションを調べる

パート 5：スイッチを SSH アクセス用に設定する

パート 6：スイッチの CLI からの SSH

背景/シナリオ

かっては、ネットワーク デバイスをリモートで設定するために使用される最も一般的なネットワーク プロトコルは Telnet でした。しかし、Telnet のようなプロトコルでは、クライアント/サーバ間の情報の認証や暗号化ができません。このため、ネットワーク スニファによるパスワードや設定情報の傍受を許す結果となります。

Secure Shell（SSH; セキュア シェル）は、ルータや他のネットワーク デバイスへの安全なターミナル エミュレーション接続を確立するネットワーク プロトコルです。SSH では、ネットワーク リンクを通過するすべての情報が暗号化され、リモート コンピュータの認証が提供されます。SSH は、ネットワーク専門家が選択するリモート ログイン ツールとして、急速に Telnet に取って代わりつつあります。SSH は、リモート デバイスへのログインとコマンドの実行に特によく使用されますが、関連する SFTP（Secure FTP）プロトコルや SCP（Secure Copy）プロトコルを使用してファイルを転送することもできます。

SSH が機能するためには、SSH をサポートするように通信側のネットワーク デバイスを設定する必要があります。この実習では、ルータで SSH サーバを有効にしてから、SSH クライアントがインストールされている PC を使用して、そのルータに接続します。ローカル ネットワークで使用する場合、一般的にはイーサネットと IP を使用して接続が確立されます。

この実習では、SSH 接続を受け入れるようにルータを設定し、Wireshark を使用して Telnet および SSH セッションをキャプチャして確認します。これは SSH による暗号化の重要性を示すものです。また、SSH 接続用にスイッチを自力で設定することにも挑戦してもらいます。

注: CCNA 実習で使用するルータは、Cisco IOS Release 15.2(4)M3(universalk9 イメージ)を搭載した Cisco 1941 Integrated Services Router(ISR)です。また、使用するスイッチは、Cisco IOS Release 15.0(2)(lanbasek9 イメージ)を搭載した Cisco Catalyst 2960 です。他のルータ、スイッチ、および Cisco IOS バージョンを使用することもできます。モデルと Cisco IOS バージョンによっては、使用できるコマンドと生成される出力が、実習とは異なる場合があります。正しいインターフェイス ID については、この実習の最後にあるルータ インターフェイスの要約表を参照してください。

注: ルータとスイッチが消去され、スタートアップ コンフィギュレーションがないことを確認してください。不明な場合は、インストラクタに相談してください。

実習に必要なリソースや機器

- ルータ 1 台(Cisco IOS Release 15.2(4)M3 ユニバーサル イメージまたは同等イメージを搭載した Cisco 1941)
- スイッチ 1 台(Cisco IOS リリース 15.0(2) の lanbasek9 イメージを搭載した Cisco 2960 または同等機器)
- PC 1 台(Tera Term や Wireshark などのターミナル エミュレーション プログラムがインストールされている Windows 7、Vista、または XP 搭載 PC)
- コンソール ポート経由で Cisco IOS デバイスを設定するためのコンソール ケーブル
- トポロジで指定されているイーサネット ケーブル

パート 1: デバイスの基本設定を構成する

パート 1 では、ネットワーク トポロジを設定し、ルータのインターフェイス IP アドレス、デバイス アクセス、パスワードなどの基本設定を構成します。

手順 1: トポロジに示すようにネットワークを配線する。

手順 2: ルータとスイッチを初期化してリロードする。

手順 3: ルータを設定する。

a. ルータにコンソール接続し、特権 EXEC モードを有効にします。

b. コンフィギュレーション モードに切り替えます。

c. DNS lookup を無効にして、誤って入力されたコマンドをルータがホスト名として変換することを防ぎます。

d. 特権 EXEC の暗号化パスワードとして **class** を割り当てます。

e. コンソール パスワードとして **cisco** を割り当て、ログインを有効にします。

f. vty パスワードとして **cisco** を割り当て、ログインを有効にします。

g. プレーン テキスト パスワードを暗号化します。

h. 不正アクセスが禁止されているデバイスへのアクセスを行うユーザに警告するバナーを作成します。

i. アドレッシング テーブルに含まれる情報を使用して、ルータの G0/1 インターフェイスを設定し、アクティブ化します。

j. 実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに保存します。

手順 4:　PC-A を設定する。

a. PC-A の IP アドレスとサブネット マスクを設定します。

b. PC-A のデフォルト ゲートウェイを設定します。

手順 5:　ネットワーク接続を確認する。

PC-A から R1 へ ping を実行してください。ping が失敗した場合、接続のトラブルシューティングを行います。

パート 2:　SSH アクセス用にルータを設定します。

Telnet を使用したネットワーク デバイスへの接続には、セキュリティ上のリスクが伴います。すべての情報がクリア テキスト形式で送信されるためです。SSH は、セッション データを暗号化し、デバイス認証の機能を提供します。そのため、リモート接続では SSH が推奨されます。パート 2 では、vty 回線で SSH 接続を受け入れるようにルータを設定します。

手順 1:　デバイス認証を設定します。

デバイス名とドメインは、暗号キーの生成時に暗号キーの一部として使用されます。そのため、これらの名前は、**crypto key** コマンドの発行前に入力する必要があります。

a. デバイス名を設定します。

```
Router(config)# hostname R1
```

b. デバイスのドメインを設定します。

```
R1(config)# ip domain-name ccna-lab.com
```

手順 2:　暗号キーの方式を設定します。

```
R1(config)# crypto key generate rsa modulus 1024
The name for the keys will be: R1.ccna-lab.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

R1(config)#
*Jan 28 21:09:29.867: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

手順 3:　ローカル データベースのユーザ名を設定します。

```
R1(config)# username admin privilege 15 secret adminpass
R1(config)#
*Feb  6 23:24:43.971: End->Password:QHjxdsVkjtoP7VxKIcPsLdTiMIvyLkyjT1HbmYxZigc
R1(config)#
```

注: 特権レベルを 15 にすると、そのユーザには管理者権限が与えられます。

手順 4:　vty 回線で SSH を有効にします。

a. **transport input** コマンドを使用してインバウンド vty 回線で Telnet と SSH を有効にします。

```
R1(config)# line vty 0 4
R1(config-line)# transport input telnet ssh
```

b. ユーザの確認にローカル データベースを使用するようにログイン方式を変更します。

```
R1(config-line)# login local
R1(config-line)# end
R1#
```

手順 5:　実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに保存します。

```
R1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R1#
```

パート 3:　Wireshark を使用して Telnet セッションを調べる

パート 3 では、Wireshark を使用して、ルータの Telnet セッションの送信データを取得および確認します。Tera Term を使用して R1 に Telnet で接続し、このルータで show run コマンドを発行します。

注: Telnet/SSH クライアント ソフトウェア パッケージが PC にインストールされていない場合は、作業を続ける前にこのパッケージをインストールする必要があります。よく知られているフリーウェアの Telnet/SSH パッケージは、Tera Term（http://download.cnet.com/Tera-Term/3000-20432_4-75766675.html）と PuTTY（www.putty.org）の 2 つです。

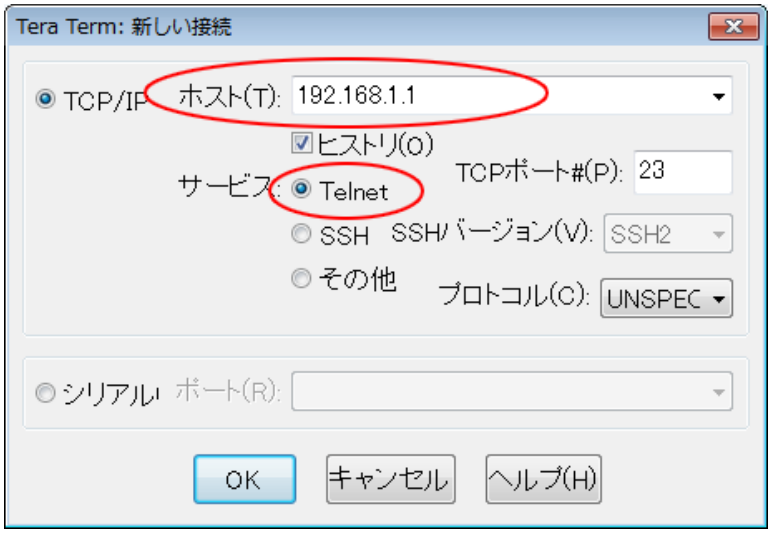
注：Windows 7 の場合、デフォルトでは Telnet をコマンド プロンプトで使用できません。コマンド プロンプトのウィンドウで Telnet を使えるようにするには、[**スタート**] > [**コントロール パネル**] > [**プログラム**] > [**プログラムと機能**] > [**Windows の機能の有効化または無効化**] をクリックします。[**Telnet クライアント**] チェックボックスをオンにし、[**OK**] をクリックします。

手順 1:　Wireshark を開き、LAN インターフェイスでデータのキャプチャを開始します。

注: LAN インターフェイスでのキャプチャを開始できない場合は、[**管理者として実行**] オプションを使用して Wireshark を開く必要があります。

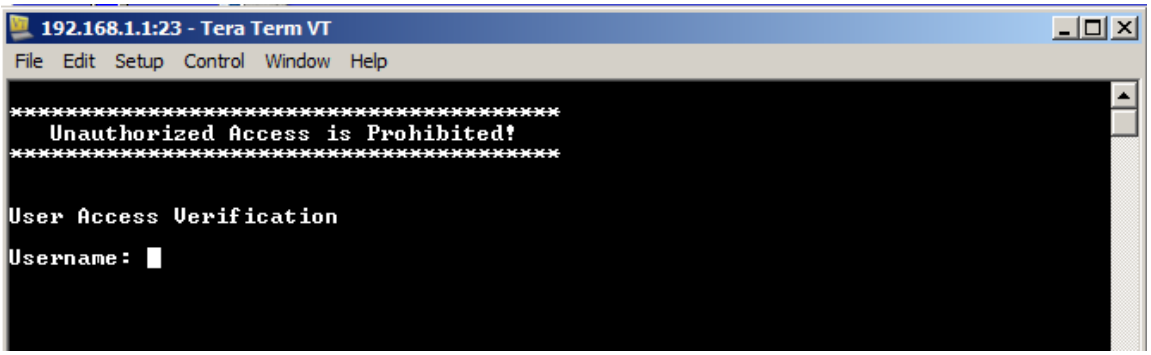
手順 2:　ルータへの Telnet セッションを開始します。

a. Tera Term を開き、**Telnet** サービスのラジオ ボタンを選択し、ホスト フィールドに「**192.168.1.1**」と入力します。



Telnet セッションのデフォルトの TCP ポートは何番ポートですか。

b. ユーザ名のプロンプトの後には「**admin**」、パスワードのプロンプトの後には「**adminpass**」と入力します。これらのプロンプトが生成されるのは、**login local** コマンドによってローカル データベースを使用するように vty 回線を設定したためです。



c. **show run** コマンドを発行します。

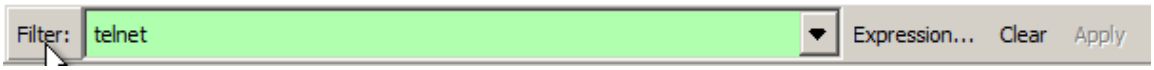
```
R1# show run
```

d. Telnet セッションを終了して Tera Term を閉じるには、「**exit**」と入力します。

```
R1# exit
```

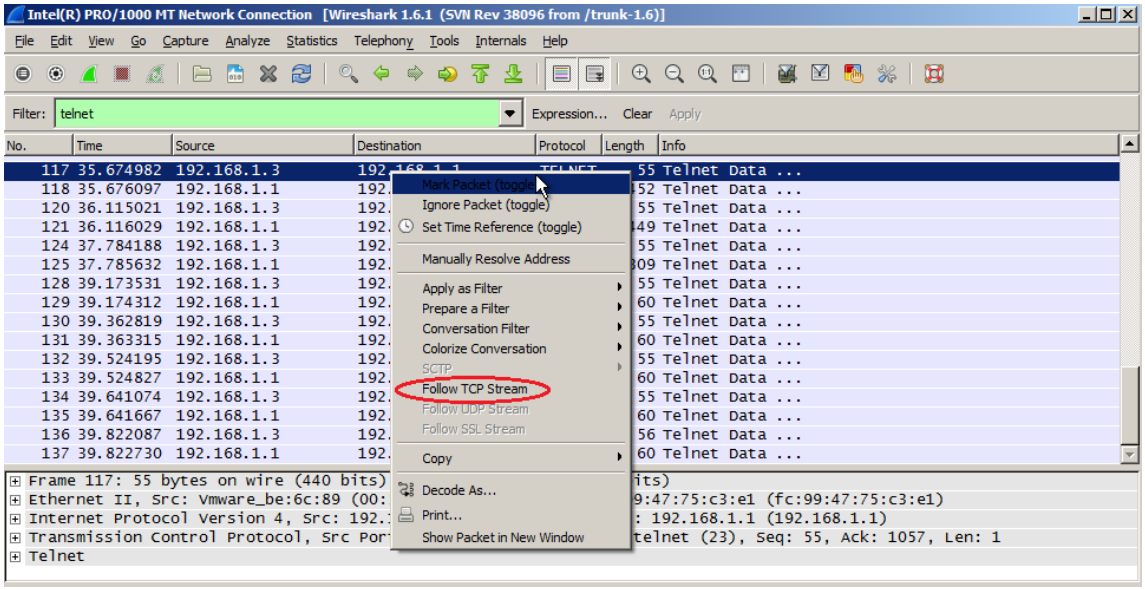
手順 3:　Wireshark のキャプチャを停止します。

手順 4:　Wireshark のキャプチャ データに対して Telnet のフィルタを適用します。

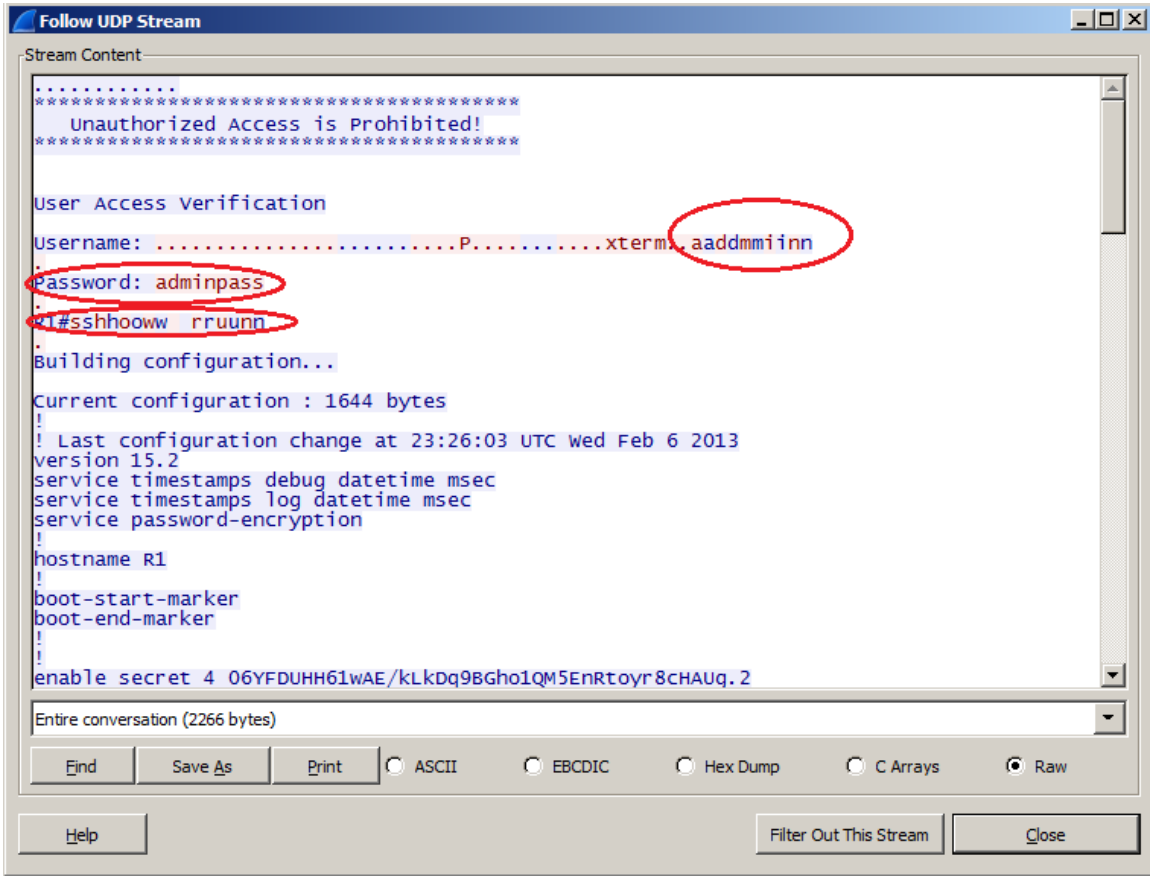


手順 5:　Wireshark の Follow TCP Stream 機能を使用して Telnet セッションを表示します。

a. Wireshark の**パケット リスト** セクションにある **Telnet** の行の 1 つを右クリックし、ドロップダウン リストの [**Follow TCP Stream**] を選択します。



b. [Follow TCP Stream] ウィンドウに、このルータによる Telnet セッションのデータが表示されます。パスワードを含むセッション全体がクリア テキストで表示されます。入力したユーザ名と **show run** コマンドの各文字が重複して表示されていることに注意してください。これは、入力した文字を画面上で確認できるようにする、Telnet の echo 設定によるものです。



c. [**Follow TCP Stream**] ウィンドウで Telnet セッションを確認し終えたら、[**Close**] をクリックします。

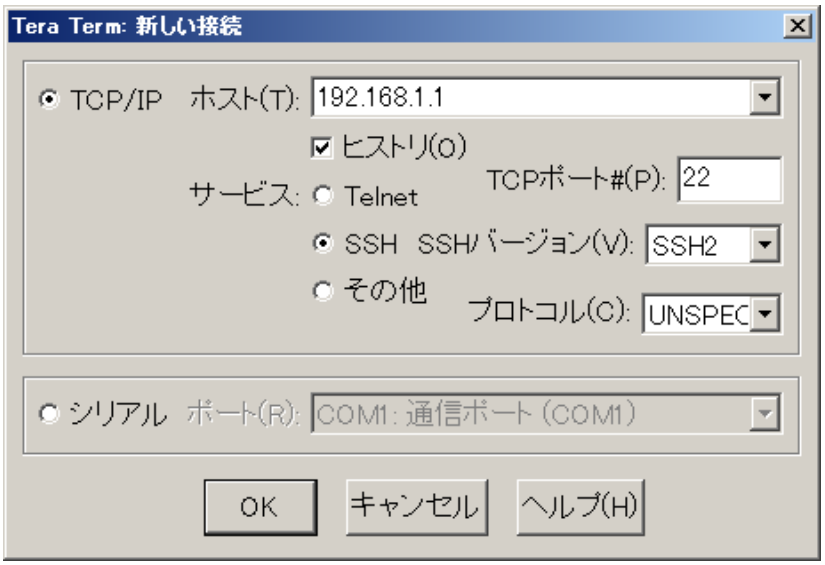
パート 4: Wireshark を使用して SSH セッションを調べる

パート 4 では、Tera Term ソフトウェアを使用して、ルータとの SSH セッションを確立します。この SSH セッションのデータのキャプチャと確認には、Wireshark を使用します。

手順 1: Wireshark を開き、LAN インターフェイスでデータのキャプチャを開始します。

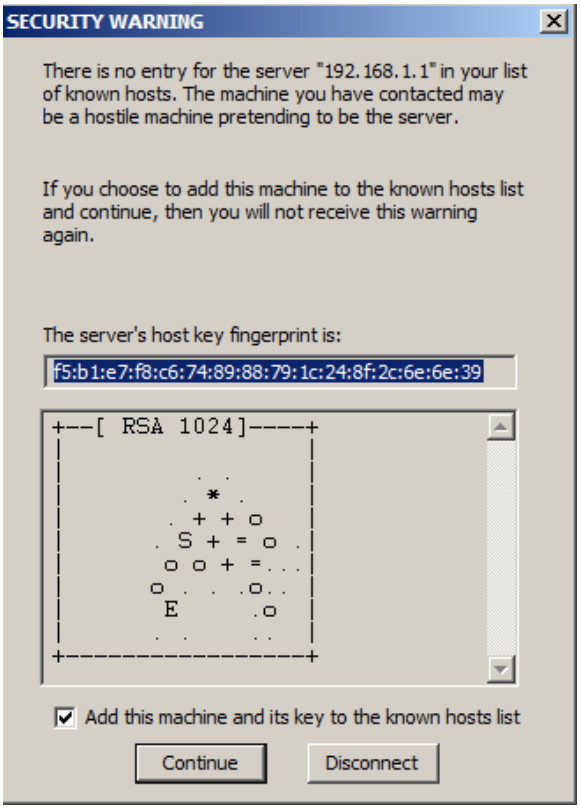
手順 2: ルータ上で SSH セッションを開始します。

a. Tera Term を開き、[Tera Term: New Connection（Tera Term: 新しい接続）] ウィンドウの [Host（ホスト）] フィールドに R1 の G0/1 インターフェイスの IP アドレスを入力します。[**SSH**] ラジオ ボタンがオンになっていることを確認し、[**OK**] をクリックするとルータへの接続が行われます。

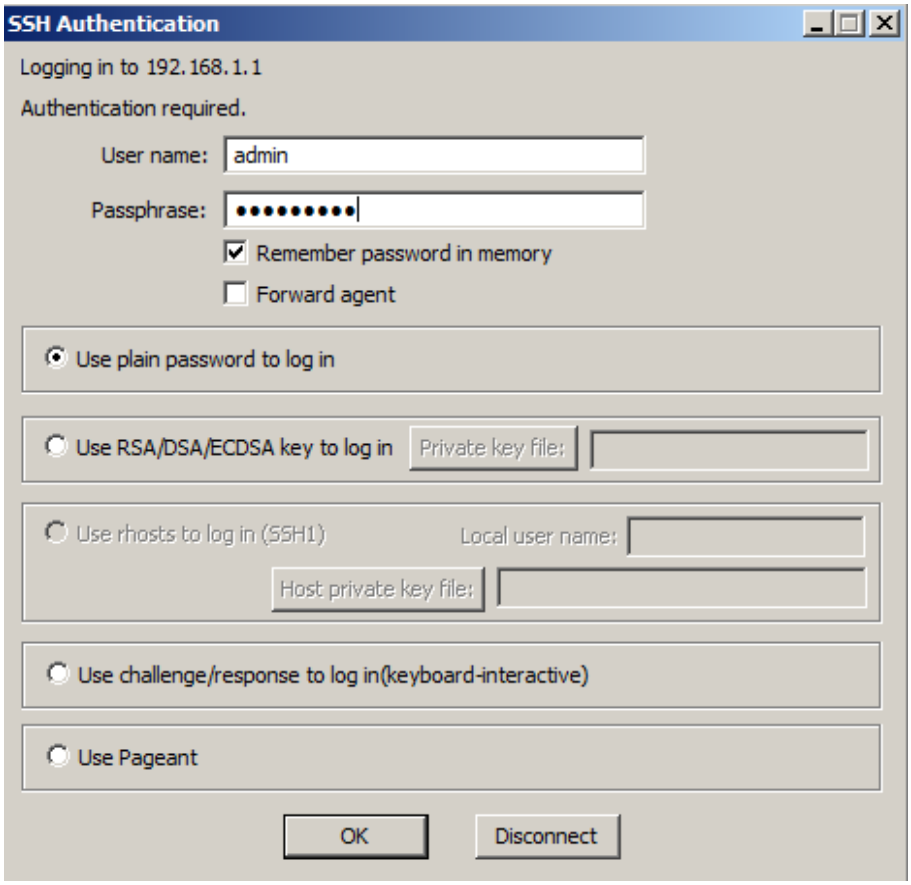


SSH セッションのデフォルトの TCP ポートは何番ポートですか。

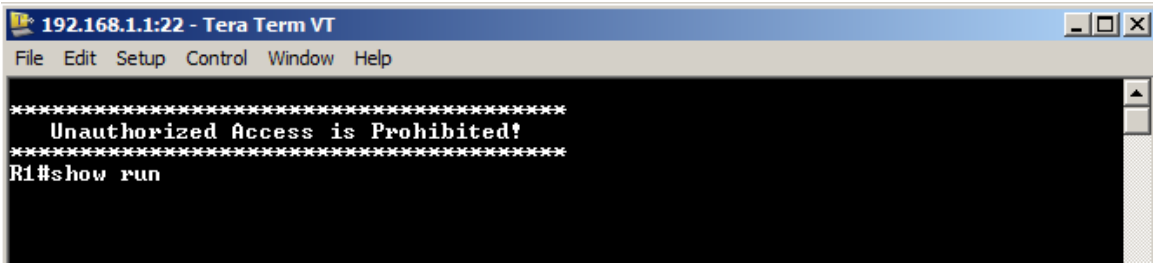
b. デバイスへの SSH セッションを初めて確立する際には、そのデバイスにそれまで接続したことがないことを通知する**セキュリティ警告**が生成されます。このメッセージは認証プロセスの一部です。セキュリティ警告を確認し、[**Continue（続行）**] をクリックします。



c. [SSH Authentication（SSH 認証）] ウィンドウで、ユーザ名には「**admin**」、パスフレーズには「**adminpass**」と入力します。[**OK**] をクリックしてルータにログインします。



d. ルータ上で SSH セッションが確立されました。Tera Term ソフトウェアの外観はコマンド ウィンドウに非常によく似ています。コマンド プロンプトで、**show run** コマンドを発行します。



e. SSH セッションを終了して Tera Term を閉じるには、**exit** コマンドを発行します。

```
R1# exit
```

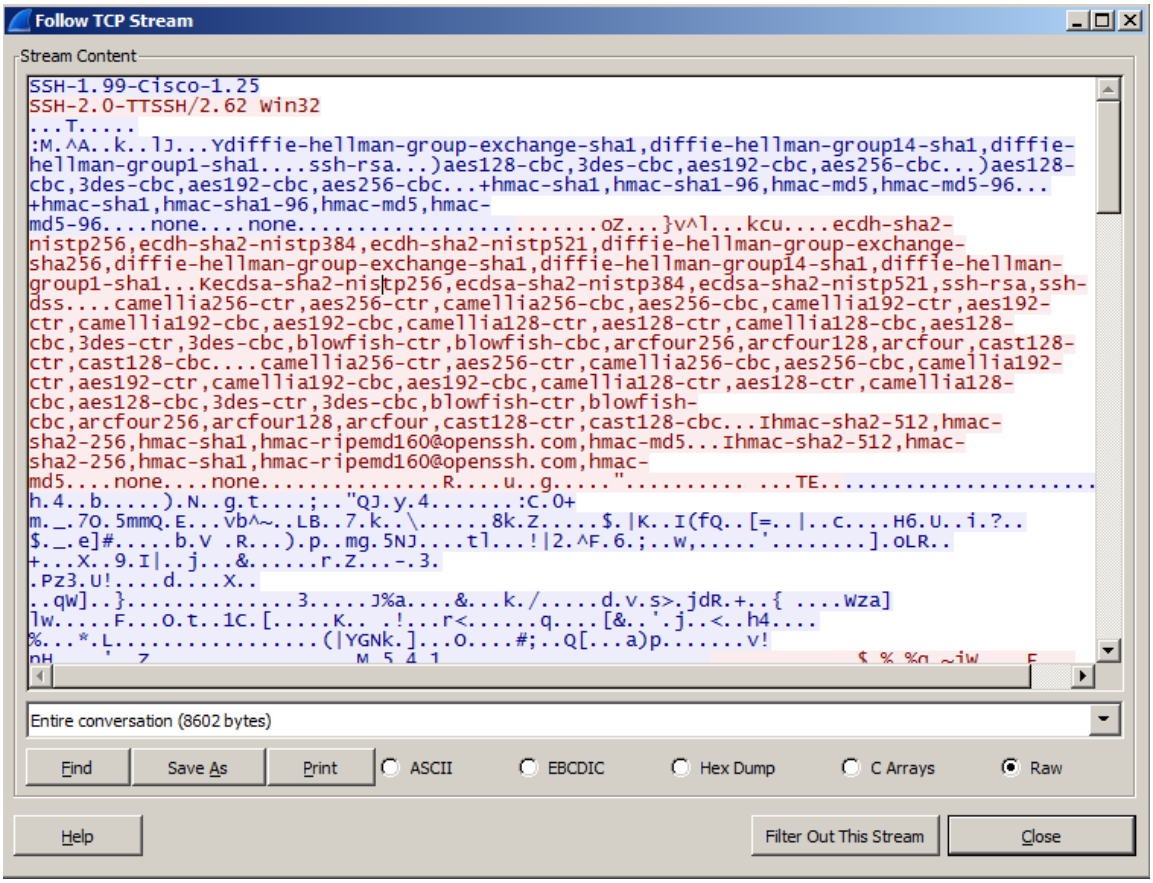
手順 3: Wireshark のキャプチャを停止します。

手順 4:　Wireshark のキャプチャ データに対して SSH のフィルタを適用します。

手順 5:　Wireshark の Follow TCP Stream 機能を使用して Telnet セッションを表示します。

a. Wireshark の**パケット リスト** セクションにある **SSHv2** の行の 1 つを右クリックし、ドロップダウン リストの [**Follow TCP Stream**] オプションを選択します。

b. [**Follow TCP Stream**] ウィンドウで SSH セッションを確認します。データは暗号化され、読み取り不可能です。SSH セッションのデータを Telnet セッションのデータと比較します。



リモート接続で SSH が Telnet よりも好ましいのはなぜですか。

c. SSH セッションの確認が済んだら、[**Close**] をクリックします。

d. Wireshark を閉じます。

パート 5:　スイッチを SSH アクセス用に設定する

パート 5 では、SSH 接続を受け入れるようにトポロジ内のルータを設定します。スイッチの設定が完了したら、Tera Term を使用してそのスイッチで SSH セッションを確立します。

手順 1:　スイッチで基本設定を行います。

手順 2:　スイッチを SSH 接続用に設定します。

パート 2 のルータでの SSH 設定のときと同じコマンドを使用して、スイッチで SSH の設定を行います。

手順 3:　スイッチへの SSH 接続を確立します。

PC-A から Tera Term を起動し、S1 の SVI インターフェイスに SSH で接続します。

手順 4:　必要に応じて、トラブルシューティングを行います。

このスイッチとの SSH セッションを確立できますか。

パート 6:　スイッチの CLI からの SSH

SSH クライアントは Cisco IOS に組み込まれており、CLI から実行できます。パート 6 では、スイッチの CLI からルータに SSH で接続します。

手順 1:　Cisco IOS の SSH クライアントで使用できるパラメータを確認します。

ssh コマンドで使用できるパラメータ オプションを表示するには、疑問符（**?**）を使用します。

```
S1# ssh ?
  -c    Select encryption algorithm
  -l    Log in using this user name
  -m    Select HMAC algorithm
  -o    Specify options
  -p    Connect to this port
  -v    Specify SSH Protocol Version
  -vrf  Specify vrf name
  WORD  IP address or hostname of a remote system
```

手順 2:　S1 からルータ R1 に SSH で接続します。

a. R1 への SSH 接続時には - **l admin** オプションを使用する必要があります。これにより、ユーザ **admin** としてログインできます。パスワードの入力を求められたら「**adminpass**」と入力します。

```
S1# ssh -l admin 192.168.1.1
パスワード:
**********************************************
  Warning: Unauthorized Access is Prohibited!
**********************************************
```

```
R1#
```

b. **Ctrl+Shift+6** キーを押すと、R1 への SSH セッションを閉じることなく S1 に戻ることができます。**Ctrl+Shift+6** キーを離して **x** キーを押します。スイッチの特権 EXEC プロンプトが表示されるはずです。

```
R1#
S1#
```

c. R1 での SSH セッションに戻るには、何も入力していない CLI 行で Enter キーを押します。もう一度 Enter キーを押さないと、ルータの CLI プロンプトが表示されない場合があります。

```
S1#
[Resuming connection 1 to 192.168.1.1 ... ]

R1#
```

d. R1 での SSH セッションを終了するには、ルータ プロンプトで「**exit**」と入力します。

```
R1# exit

[Connection to 192.168.1.1 closed by foreign host]
S1#
```

CLI からサポートされる SSH のバージョンはいくつですか。

復習

それぞれに個別のユーザ名を持つ複数のユーザが、同じネットワーク デバイスにアクセスできるようにするにはどうすればよいですか。

ルータ インターフェイスの要約表

ルータ インターフェイスの要約				
ルータのモデル	**イーサネット インターフェイス #1**	**イーサネット インターフェイス #2**	**シリアル インターフェイス #1**	**シリアル インターフェイス #2**
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

注: ルータがどのように設定されているかを確認するには、インターフェイスを調べ、ルータの種類とルータが持つインターフェイスの数を識別します。各ルータ クラスの設定のすべての組み合わせを効果的に示す方法はありません。この表には、デバイスにイーサネットおよびシリアル インターフェイスの取り得る組み合わせに対する ID が記されています。その他のタイプのインターフェイスは、たとえ特定のルータに含まれている可能性があるものであっても、表には一切含まれていません。ISDN BRI インターフェイスはその一例です。カッコ内の文字列は、インターフェイスを表すために Cisco IOS コマンドで使用できる正規の省略形です。